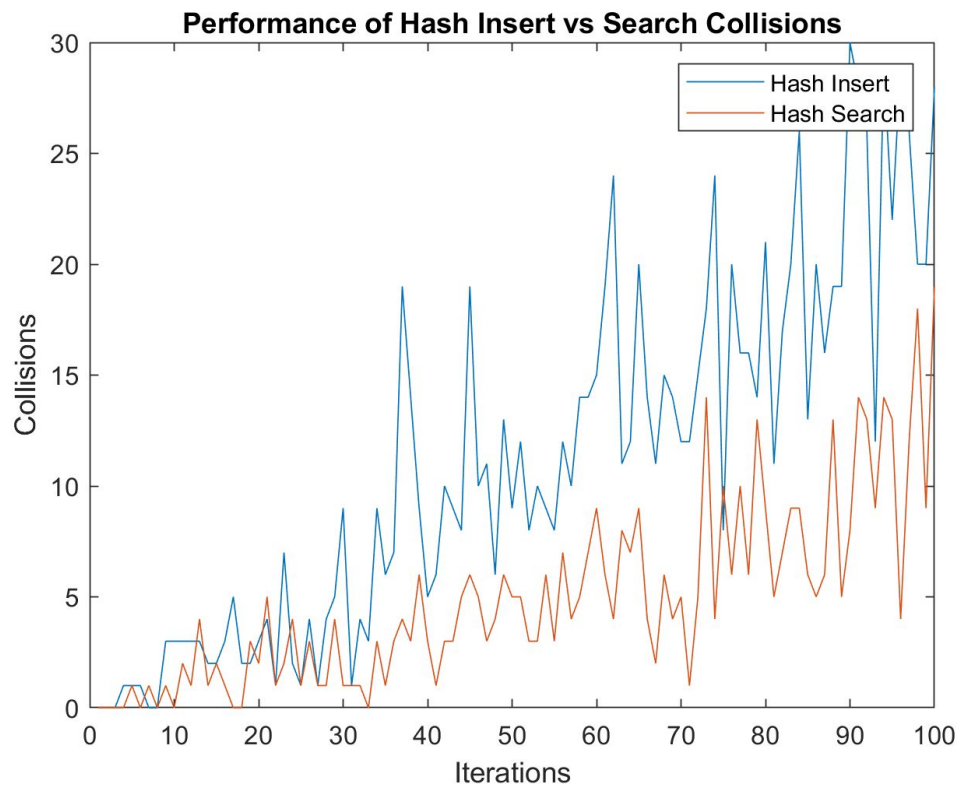
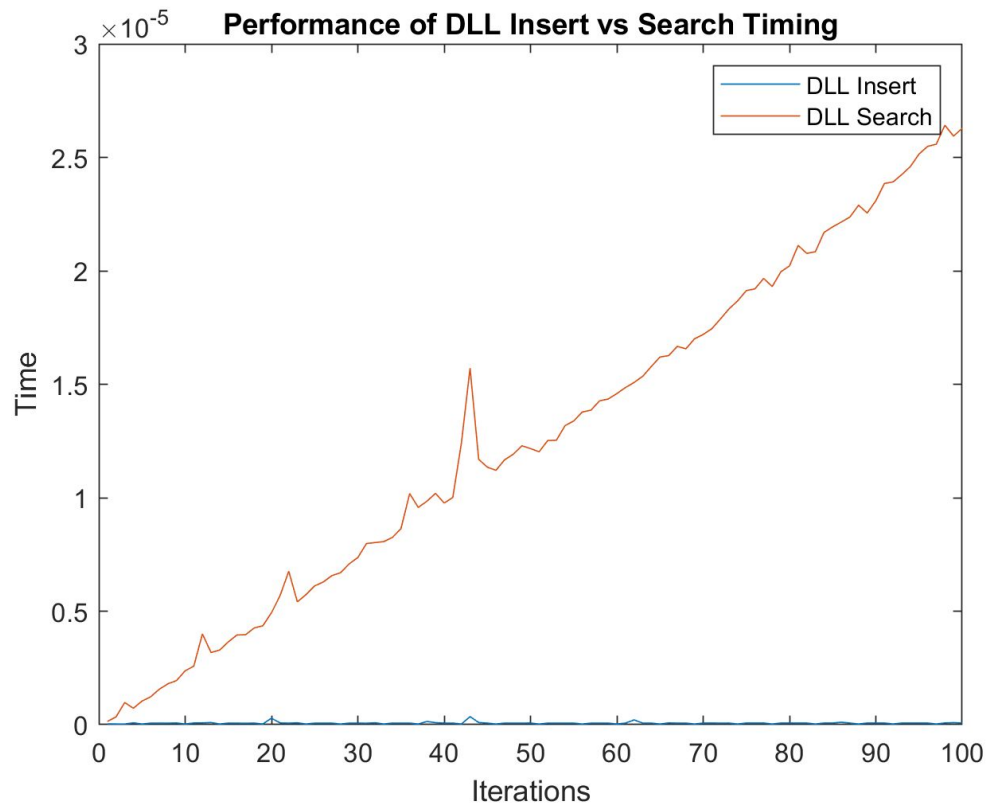
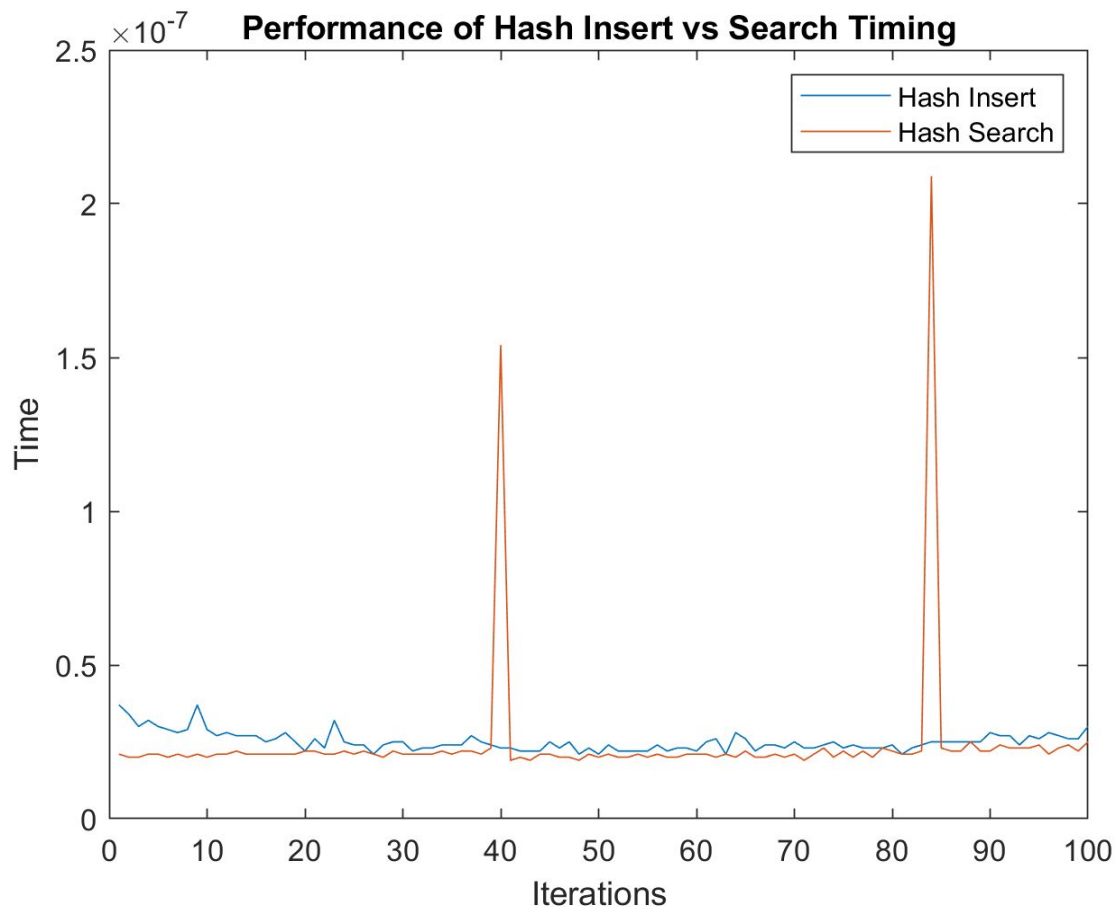
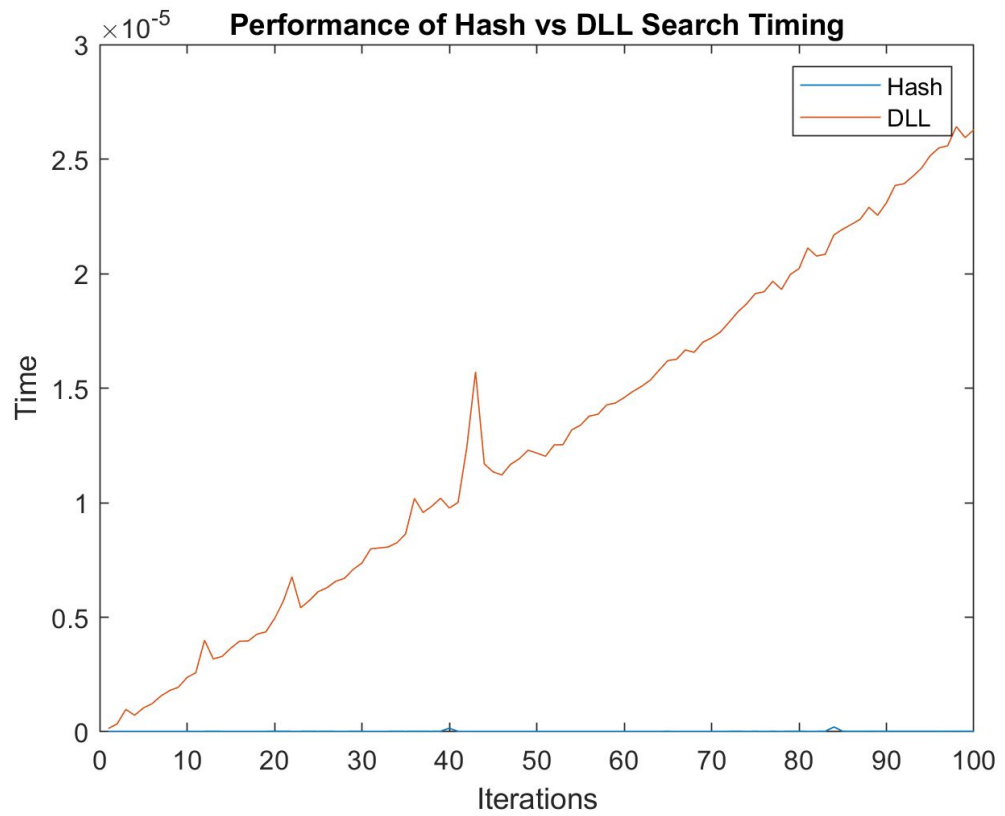
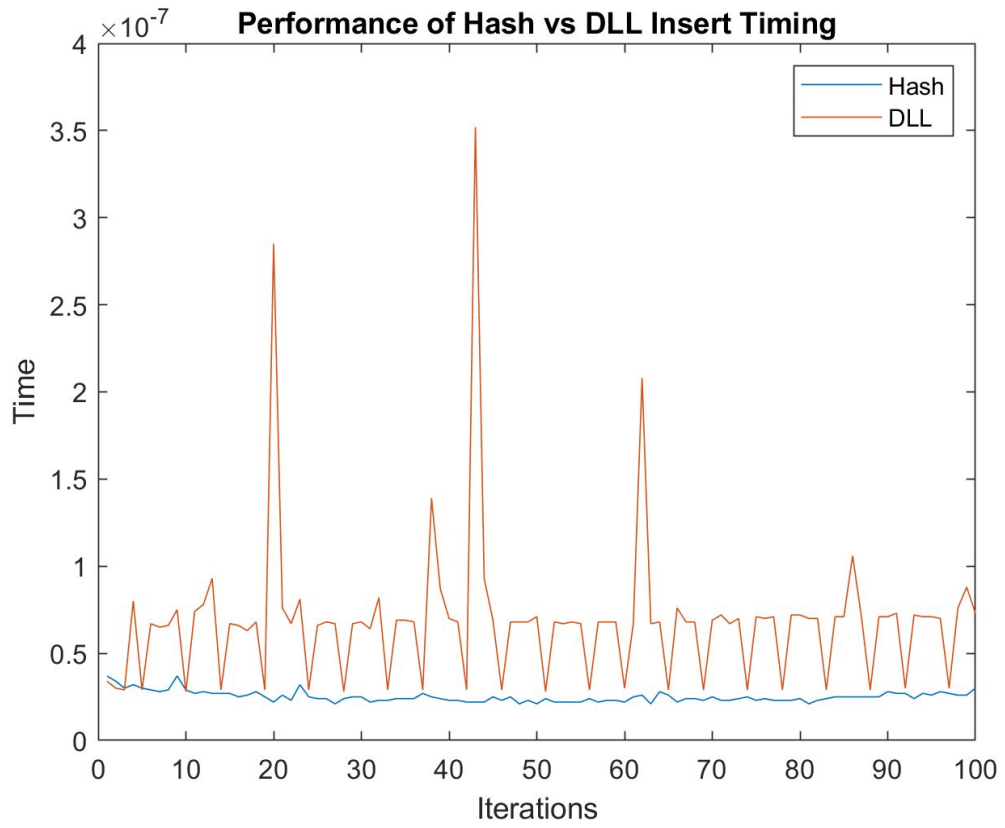


Medical Tracker

From the graphs, I think it would be safe to conclude that the hash table using quadratic probing would be the best data structure for the Medical Tracker Company. Based on the graphs we can see that the Doubly Linked List and the Hash Table had relatively similar data when it came to inserting the data, however, we can see that the hash table was about about 2 to 3 times more efficient than the doubly linked list. This could be due to the fact that the hash table's search times were extremely low compared to the double linked list and they barely showed up on the comparison graph below. As we look further into the graphed data, we can also see that the doubly linked list had a relatively constant growth in terms of search time while the hash table was constant all the way through the 100 iterations and the search time did not increase almost at all due to the quadratic probing nature of the algorithm. It is because of this that the data has less collisions frequently which means that the search is almost instantaneous. In comparison, the doubly linked list has to check every key inside the linked list so it would need to traverse the complete DLL to find the corresponding key. It is because of this that the hash table would be the better data structure since it would be a lot faster in terms of overall speed due to the nature of quadratic probing which results in almost instantaneous searches rather than traversing the full list of the DLL.







In terms of the bubble sort and the heap sort algorithms, based on the graphs below, it seems that the heap sort is vastly more efficient and fast compared to the bubble sort. This may be because the bubble sort will need to traverse the entire array which means that it would take much longer to sort all the values. The bubble sort works by traversing the array and comparing the first 2 values and swapping them if the second value is less than the first, then moves on and repeats the process. This continues all the way until the end of the array. Based on this reasoning, I feel that the heap sort would be much shorter in terms of time to sort the entire data set.

I believe that the heap sort was faster than the bubble sort because the way that the heap sort is able to store the data was similar to that of a tree of which each leaf would have a max of 2 children leafs compared to the bubble sort which used a long incrementing array of elements. This would make the heap sort much more efficient and faster at the sorting since it would be comparing and swapping the values of 3 different data points rather than 2 of the bubble sort. This means that position of the values once they have been swapped could be moved a lot in the tree or the array. This can be seen and proved using the graphs below because the bubble sort exponentially increases in terms of time the more values that are inserted, while the heap sort was near 0 or tenths of a millisecond in terms of time for the swapping and sorting. Therefore, from the graphs, I think it would be safe to conclude that the heap sort algorithm would be the best data structure for the Medical Tracker Company.

